Screencasts: enhancing coursework feedback for game programming students revisited

Law, Bobby

Published in:
International Journal on Advances in Software

Publication date:
2018

Document Version
Author accepted manuscript

Screencasts: Enhancing Coursework Feedback for Game Programming Students Revisited

Robert Law

School of Computing, Engineering and Built Environment
Glasgow Caledonian University
Glasgow, Scotland
Email: robert.law@gcu.ac.uk

***Abstract*—Feedback is an important part of learning and, as such is vital for students to develop and progress throughout their academic life. Programming can be an abstract concept that students find challenging to comprehend therefore good feedback is important to their progress and their motivation to continue programming. This paper will discuss the process of enhancing coursework feedback for Game Programming students through the use of screencasts. The hypothesis being that game programming by its nature is audio-visual thus, providing feedback using an audio-visual medium should increase the student's perception of their feedback such that it is perceived to be clearer, easier to comprehend and personalised.**

***Keywords*—Screencasts; Feedback; Software Development.**

I. Introduction

Following on from work done by Law [1]: this paper revisits the concept of enhancing coursework feedback for Game Programming students through the use of screencasts with a view to offering a template that can be utilised in the production of screencasts, which both minimise the Lecturer's work load and maximises the students feedback.

The United Kingdom's (UK) National Student Survey (NSS) [2] is a survey for final year students at all of the UK's publicly funded Higher Education Institutions (HEIs) and is administered by Ipsos MORI. The NSS comprises of 27 questions across eight categories attempting to capture the students learning experience. The NSS acts as a barometer of student satisfaction and thus, is an influential survey giving the student body a collective voice. The data from the survey is publicly available and is used by prospective students when choosing their potential University.

This survey has a number of different sections, one of, which is Assessment and Feedback. The perennial view from students suggests that there is scope for improvement with regard to Feedback. Comparing all eight categories it can be seen that Assessment and Feedback is continually at the bottom. This would suggest that there is still room for improvement. Table I shows all the sections of the questionnaire and their corresponding percentage satisfaction rating. It is noticeable, from Table I, that satisfaction with Assessment and Feedback is between 5 and 14 percentage points behind 7 of the 8 remaining categories suggesting that the students'impression of feedback and the instrument of feedback delivery have not met entirely with the students'expectations [3], [4].

Viewing the statistics on a nation by nation basis against the UK average creates an interesting picture of how students in each of the four nations differ in their perceptions of the level of feedback they receive. Figure 1 shows a comparison of all four nations. Working in an academic institution in Scotland the picture painted is somewhat alarming with Scotland six points below the UK average [5]. The Assessment and Feedback section of the survey is comprised of five questions; two relating to assessment and three relating to feedback. The feedback questions are shown in Table II. The questions in Table II emphasize the students'desire for expeditious, clear and detailed feedback [6].

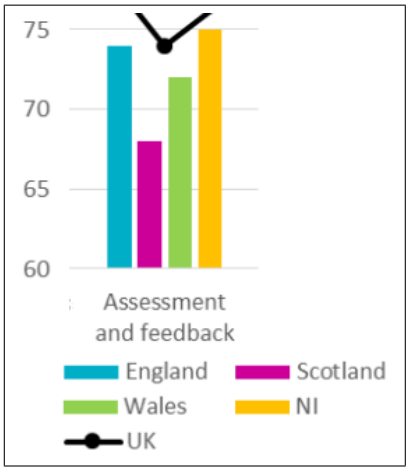

Figure 1. Assessment and Feedback results 2016 by nation

TABLE I. Percentage satisfaction across categories from NSS questionnaire

Categories	*2015*	*2016*
The teaching on my course	87	87
Assessment and feedback	73	74
Academic support	82	82
Organisation and management	79	79
Learning resources	85	86
Personal development	83	82
Overall satisfaction	86	86

The remainder of this paper is organized as follows: Section II will provide an overview of the author's rationale for the use of screencasts within the feedback process; indicating the nature of the cohort and the subject area studied. Section III will provide information about pedagogical issues related to screencasting, Section IV offers an introduction to the

TABLE II. EXTRACT OF FEEDBACK QUESTIONS FROM NSS QUESTIONNAIRE

Feedback Questions asked as part of NSS
Feedback on my work has been prompt.
I have received detailed comments on my work.
Feedback on my work has helped me clarify things I did not understand.

technologies available for screencasting. Section V presents an overview of the screencasting process, while Section VI reflects on the informally gathered feedback from the student cohort. Section VII discusses issues encountered by the author during the creation of the screencasts. Section VIII attempts to derive conclusions based on the synthesis of Sections III, IV, VI and VII. Section IX offers ideas for future work.

II. RATIONALE

Teaching programming, and in particular, game programming it can be difficult to offer students feedback on coursework submissions that are not either too generic and brief or ultimately too verbose and overcomplicated. Getting the balance of written feedback correct can be a daunting task. Thompson and Lee [7] suggest that feedback is "a pedagogical tool to improve learning by motivating students to rethink and rework their ideas rather than simply proofread and edit for errors." . Interestingly, Thompson and Lee [7] quote Notar, Wilson and Ross that "feedback should focus on improving the skills needed for the construction of end products more than on the end products themselves". This particular observation is very apt for teaching programming concepts and programming languages as the feedback given is in the context of the students programming skills rather than their end product, in this case their game. The feedback is intended to improve the students ability to produce structured, economical code and illustrate the necessary skills for debugging program code.

The author teaches game programming modules at various levels within the undergraduate programme BSc (Honours) Game Software Development. It would seem natural for game programming students who primarily work with a very audio visual medium to receive feedback for their programming coursework as an audio-visual screencast. It was therefore decided to implement a trial with a second year cohort undertaking the module Game Programming 1. This module was chosen as it was a core module for both the Game Software Development students and the Game Design students. The module introduces students to coding using C++ and OpenGL with the emphasise on the production of a 2D game prototype. The module had approximately 70 students participating in it with a near even split of Game Software Development and Game Design students.

The coursework required the students to create a game of their choosing. The coursework specification provided the students with a number of requirements that had to be met and a marking scheme was provided as a guide to the aesthetic appearance of their game and the functional aspects of the underlying code.

This paper will focus on post coursework feedback, which, in this case represents feedback given to the student after completion of the module. The submission date for the coursework is normally the last week of term, therefore, feedback would normally be provided in a written format, distributed via email.

The aim of the research is twofold: to better understand the delivery of feedback to students undertaking programming courses via the medium of screencasts such that the students feel that they have gained a meaningful commentary on their coursework submission, which will, hopefully, lead them to improve their subsequent submissions; and to identify a process or template that can be used by Lecturer's to minimise their work load while maximising the amount of feedback given to the student.

III. PEDAGOGICAL ISSUES

So what is a screencast? For the purposes of this paper a screencast will be defined as a recording of the current content of the computer screen with an audio narration providing relevant commentary, i.e., feedback [8], [9], [10]. For this reason Atfield-cutts [11] suggests that " ... video feedback potentially, such a powerful enabler for programming students in particular." As part of their learning it is important for students to receive feedback on any of the work that they produce. However, Atfield-cutts [11] identifies that "Student engagement with feedback is often lacking and in that case, a valuable learning opportunity is missed.", thus, it is important to find ways in, which, students can be encouraged to be part of the feedback loop.

It is postulated by Thompson and Lee [7] that student reluctance to engage with the feedback process maybe due to an attempt to create an equilibrium between study, home and work life, hence, Atfield-cutts [11] suggests that, in order to re-engage the students with the feedback process, the process itself must be perceived by the students to require less time and /or effort, or it must be deemed more pleasant and/or useful by students.

Race [12] identifies a number of common formats used to disseminate feedback to students: handwritten, word processed, model answers/solutions, rubric proformas, oral feedback, email and computer marked assessment. These methods can be issued individually or as general feedback based on the performance of a cohort or group.

Race [12] suggests five attributes of feedback: Timely, intimate and individual, empowering, open doors not close them and manageable. Timely feedback is a goal that is highly desired and greatly prized, but, can be dictated by class size or other commitments. Intimate and individual feedback should reflect the student's own submission. Empowering feedback is harder to achieve, as it is a balancing act between positive feedback and a critic, warts and all, of the student's submission. Open doors, not close them refers to the use of language within feedback and the expectation this can set for the student and the feedback they receive for their next submission. Manageable, is viewed from the perspective of both the student and the lecturer, i.e., the effort expended by the lecturer to produce the feedback and the volume of feedback received by the student could cause them to miss something important [12].

Using the written word to provide annotated feedback to students can be taken out of context [10] and therefore the benefit of the feedback can be lost. Worse still, the feedback taken out of context can be misconstrued as a criticism of their work [9] rather than a pointer to improvement. The loss of visual and aural cues, which aid understanding [13], from the written feedback process is therefore something that

screencasting can help combat. Moreover, screencast feedback has the potential to "provide more information to students about their work compared to the written commentary." [14]

As part of Evans [15] "12 pragmatic actions" for effective feedback, one suggestion is for students to be presented with an early assessment opportunity such that they can receive early feedback, which, can be built upon prior to final submission.

It has been mooted that audio-visual screencasts can create for the Lecturer the concept of "social presence" and "an opportunity for conveying positive encouragement through intonation." [9]. This ability to use intonation to emphasize important [3] aspects of feedback make the use of screencasts a benefit for the student. Indeed, Seror [16], believes that screencast feedback offers the ability to provide "a more conversational and personal form of feedback." Couple this with the ability to hear the feedback in the manner the Lecturer intended it and the loss of the visual and aural cue associated with face to face feedback are somewhat restored. The volume of information that can be presented to the student via the audio aspect of screencasts is far larger than written feedback alone and in a shorter time period [3], [17], [18].

Galanos et al. cite the use of screencasts as a method of giving a student personalised feedback by recording the lecturer debugging the students program code while commenting on it [19]. Also suggested is the use of an attached webcam to offer "picture in picture" of the lecturer while debugging the program code, helping to offer that personal touch [19].

It has been suggested that screencasts can aid the student's understanding of their feedback by negating the need for continual cross-referencing between feedback and assessment and secondly the use of conversation style feedback rather than a more formal written academic feedback [9]. It has also been suggested that students find it clearer to "understand the marker's reasoning" [8] and comments [20] when presented in a screencast.

Clarity of feedback is important to students [21]; they do not want to receive feedback that could be deemed "vague, unclear and confusing" [22]. Thus, the audio-visual nature of screencasts can help enrich the feedback pinpointing unambiguously exactly what is being commented on [22]. The promptness or timeliness of feedback is another concern for students as evidenced by the low scores in the National Student Survey [5]. Hope suggests that educators are under an "obligation to provide meaningful feedback within a reasonable timeframe" [3]. Mathisen proffers anecdotal evidence from the field that screencasts can provide more feedback and can be produced in less time [22].

It has been mooted by O'Malley that one of a quartet of criteria needed for feedback to be effective is for it to be personal [23]. Screencasting offers the student personalised feedback that is tailored to their submission [9]. Chewar and Matthews state that the use of screencasts to provide feedback allows for more detailed, accurate and robust feedback [24]. Thomson and Lee also suggest that feedback given through the use of screencasts has the capacity to motivate and boost the students engagement with their learning [7].

Sugar et al. [25] undertook to research the anatomy of a screencast with a view to developing a framework for the production of screencasts to better aid Lecturers in producing effective learning screencasts. Although this research was focused on the production of screencasts for learning e.g., how to save a spreadsheet as a CSV file, this framework offers potential for the production of feedback screencasts.

Sugar et al. [25] framework consists of two categories: Structural elements and instructional strategies. These categories are further subdivided as follows: Structural elements comprises of "bumpers, screen movement and narration"; Instructional elements comprises of "provide overview, describe procedure, present concept, elaborate content, and focus attention." Figure 2 shows the framework with a further layer of subdivision.

Examining each of the structural and instructional elements suggests that this framework could be adapted to reflect the creation of feedback screencasts. Although, the intention of feedback screencasts is to add a level of personalisation, a framework or checklist would act as a valuable guide to the desired content of a feedback screencast.

The three structural components offer a clear set of tools for adding a degree of personalisation to a feedback screencast. For example, bumpers, a term borrowed from radio broadcasts [25], is a technique used to offer a salutation and/or a valediction to the screencast. This allows the Lecturer to provide an opening and closing greeting to the student e.g., possible opening statement

> *Hi Jim, Well done on completing the coursework! I will now provide you with feedback on your submission, which will hopefully prove useful.*

and a possible closing statement

> *Jim, I have covered a number of aspects in your submission and I hope that the feedback has helped elaborate on the key aspects of the coursework and how your submission met that criteria. Thanks, Bobby.*

The examples above exemplify the type of personalisation that can be applied to the feedback screencast.

Screen movement can be split into two types: static or dynamic; static screen movement is "a constant frame in, which the cursor moves within that frame" and dynamic screen movement is "the capture frame moves around the screen, keeping the cursor in the center." [25] A mixture of both types could be used for various aspects of programming feedback, for instance, while playing the student's game static screen movement would be appropriate, but, providing feedback on the student's code would benefit from the use of dynamic screen movement allowing the Lecturer to hone in on the desired code fragment.

Narration is an important aspect of a feedback screencast as it is the Lecturer's route to personalisation. Sugar et al. [25] define narration as explicit and implicit; explicit narration depicts what can be seen on screen and implicit narration refers to more generalised commentary. A combination of both would be appropriate for a feedback screencast.

The five instructional components offer a set of tools, which can be mixed and matched were appropriate to add the necessary degree of personalisation to a feedback screencast. As noted by Sugar et al. [25], not all of these instructional components were found in instructional screencasts, therefore, not all of these components will be required in a feedback screencast.

In the context of instructional screencasts "Provide Overview" delivers the "necessary background information that learners need in order to understand the context and/or the purpose of the screencasting topic;" [25]. This approach is also feasible for feedback screencasts as it seems appropriate to indicate to the student what the assessment was designed to test. Seror [16] exemplifies this approach saying "I typically start a recording with a few brief words about what I will be focusing on and how the feedback will proceed ..."

When feeding back to a student about the particular way they have coded their game "describe procedure" allows the Lecturer the ability to take an aspect of the coursework and relate the appropriate programming technique required to satisfactorily implement it.

The Lecturer can use the "present concept" strategy to identify sections of the student's code explaining how their code could have been improved or optimised through rearranging the code segment, aligning it to a design pattern, and explaining why it is a better solution.

The concept behind "focus attention" is to use a combination of the mouse pointer and narration to draw the student's attention to an area of their code that has been implemented to a high standard or could be improved. This could also be enhanced by using dynamic screen movement.

The final instructional component "elaborate content" is an opportunity for the Lecturer to "enrich" the student's comprehension and provide the student with alternative approaches that will expand their learning [25].

Subsequently, having examined Sugar et al. [25] screencast framework for instructional screencasts, it is evident that this framework is a viable framework for use with Feedback screencasts.

IV. Technology

There are a number of different combinations of hardware and software that can be used to create a screencast. The following sections will describe the hardware and software used by the author to create feedback screencasts.

A. Hardware

To capture good quality audio it is advisable to refrain from using the built-in device microphone but instead opt for a headset or external microphone [3], [26], [27]. The benefit of using a headset is the consistent distance from the mouth [28] and the ability to position it slightly below the mouth to minimize the noise of breathing [26].

B. Software

A number of software packages are available and these range from desktop applications to web based applications, which, in turn, vary in price from free to hundreds of pounds [28].

Table III illustrates a small selection of available screencasting software including a brief description of the software, highlighting its main features, has been provided along with a web link to the software. Kilickaya [14] cautions for the need to select screencast software wisely, suggesting "A benefit-cost analysis should be conducted before making the choice." It is worth noting that free software may well suffer from limitations of functionality or may not have some of the desirable advanced features of their paid for counterparts [14].

Software used for this paper was Screencast-O-Matic a web based application offering a limited version free. The free version allows up to 15 minutes of recording, recording from screen and webcam, the ability to publish to YouTube and the ability to save in popular formats such as .MP4, .AVI and .FLV. It is relatively easy to use [10] and has a very handy countdown before recording begins.

V. Recording Screencast Feedback

Although the screencast in this instance is being created in response to an unknown entity it is still important to apply the rules of creating instructional screencasts by planning [28]. Planning is very important [29] as there will be a number of areas that will require feedback. For the game produced by the students the coursework feedback was broken into the following areas: aesthetics, game play, code structure and compilation. Each of these areas was broken down further with key points: aesthetics covered the games look and feel and interface design; game play covered the ease and enjoyableness of the game, responsiveness of game objects to keyboard/gamepad interaction; code structure covered neatness, use of the fundamental programming building blocks, use of language features, data structures, and the object oriented paradigm; compilation covered the programming compiling and the appropriate use of compilation switches.

Unlike recording a conventional educational screencast there is no need to produce a script [30] as the coursework submissions will not be predictable and a script can depersonalize the feedback and make it feel unnatural [4]. Armed with the marking scheme and the aforementioned plan the process of creating the screencast could be started. A number of considerations were taken into account before commencing the screencast process:

- Determining a location, which has a low level of background noise [26] and little chance of being interrupted.
- Use a good quality headset, positioning the microphone slightly below the mouth [26].
- Switch off any software that activates pop ups such as email, Facebook or instant messenger as these could end up being recorded [4].
- Use and stick to the devised plan for consistency.
- Speak naturally and positively [30] making good use of intonation [3].
- Use of the pause button [28] at the end of each section to allow time to gather one's thoughts prior to the start of the next section.
- Screencast duration should be between five and ten minutes [14].

Having evaluated Sugar et al. [25] screencast framework it seemed like a logical decision to incorporate the framework into the production of the feedback screencasts.

"Bumpers" were incorporated, allowing a quick introduction to the student, using their name, further using the approach of "Provide Overview" aspect of the framework, explaining the key aspects of the marking scheme being used. At the

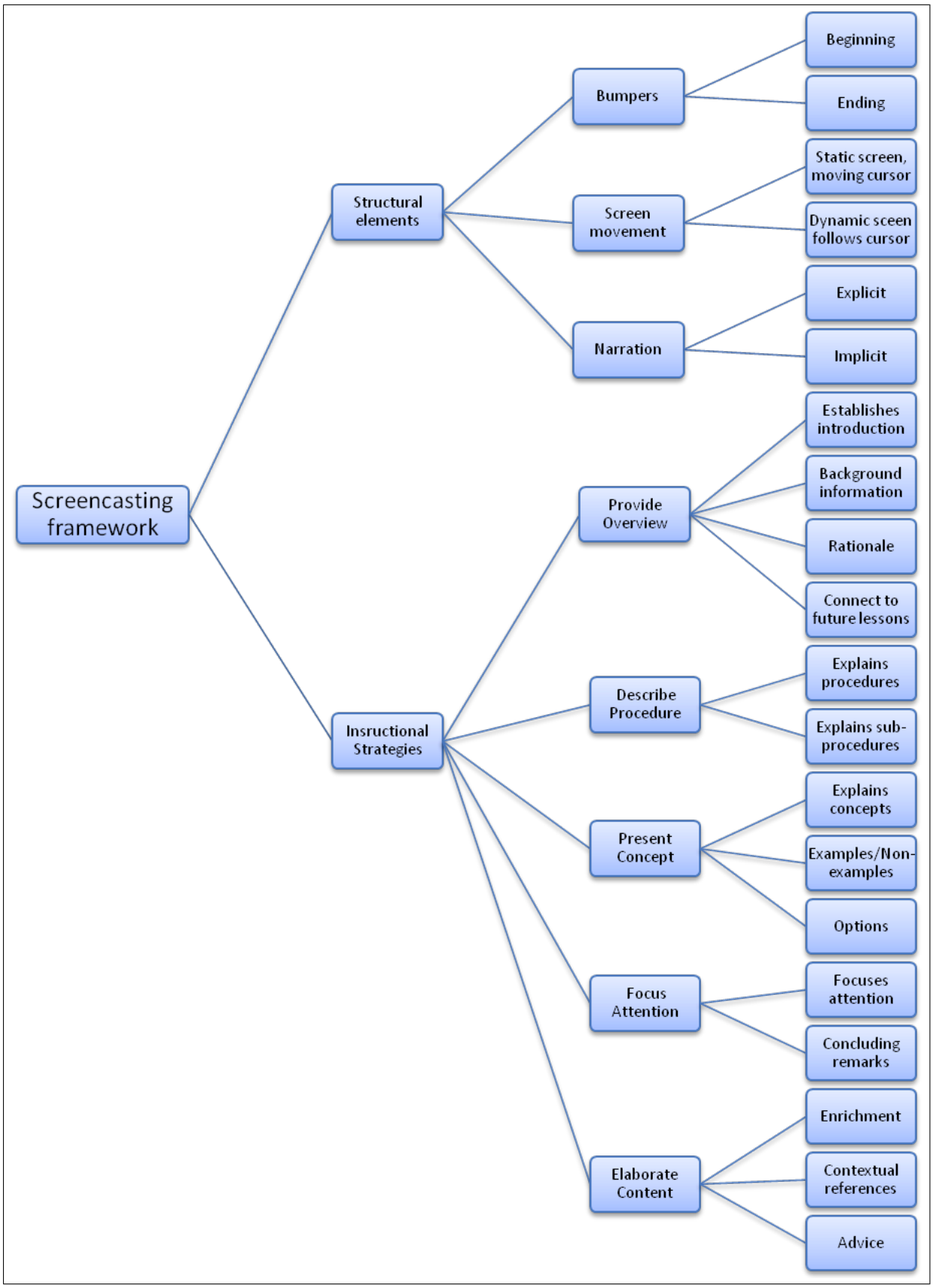

Figure 2. Sugar et al. [25] Screencast Framework

TABLE III. SELECTED SCREENCASTING SOFTWARE

Software	*Description*	*URL*
Screencast-O-Matic	Three plans: Free (Features Limited to 9 from 18), Deluxe and Premier (paid for). Hosting and Sharing available across all plans, but, with varied restrictions. Editing Tools available for Deluxe and Premier plans.	https://screencast-o-matic.com/
SnagIt	Commercial Software; Requires payment (Free Trial available); Discount for Education; Screen capture, recording and in built editor	https://www.techsmith.com/screen-capture.html
Camtasia	Commercial Software; Requires payment (Free Trial available); Discount for Education; Screen capture, Recording, video editor; can include interactive quizzes	https://www.techsmith.com/video-editor.html
Jing	Freeware; time limited and basic	https://www.techsmith.com/jing-tool.html
Screencastify	Chrome extension; Saves to Google Drive, YouTube, Export as .MP4: Mouse focus, Draw with Pen, Embed Webcam; Free and paid for version.	https://www.screencastify.com/

end of the screencast, the student, again using their name, is given a summary, the key criteria of what was being assessed, their couserwork performance and encouraged to contact the Lecturer if they would like anything explained further, which again, illustrates the concept of the "Provide Overview" aspect of the framework.

"Screen Movement" within the feedback screencasts was mainly static, but, an attempt was made to use dynamic screen movement to focus in more tightly on particular code segments and when showing the debugging process. Static screen movement is applicable when recording the students game running as the whole screen is visible. During the recording process all mouse movements and clicks are visible to the viewer as a large coloured circle that will change colour when the mouse button is clicked. This is exceptionally useful for giving the student unambiguous and precise feedback on their user interface design and layout pointing out what is considered good and what needs improving. Seror [16] outlines virtually this approach as part of a "work flow" adhered to when producing a feedback screencast; my typical work flow begins with opening the file that contains a students assignment .... record the full screen of my computer. Using a headset microphone, I then begin to read and comment on the text orally and visually. All oral comments are recorded in synch with my mouse movements as I highlight and/or edit various sections of text.

"Narration" is a key aspect to recording a useful and engaging feedback screencast and it is important to synchronise the narration with any mouse actions as this will help "Focus attention" of the student to any aspect of feedback and praise being delivered. As far as possible, every attempt was made to keep the narration explicit such that the student was left with no ambiguity about the comments being made. A tip well worth remembering when narrating a screencast is, to use the pause button on a regular basis, as this will allow time to survey and reflect on the next aspect of the marking scheme before proffering any feedback [16]. Also, as far as possible refrain from the use of implicit descriptions as this can lead to a level of ambiguity that will not be beneficial to the student. Again, during the narration it is useful to utilise the "Provide Overview" aspect of the framework to highlight to the student what is being assessed and how well they met this criteria.

The following structural aspects of the framework are not necessarily required for all students, but, are very useful for weaker students who have submitted a coursework, which has not met the desired criteria for a pass or is a borderline pass. These aspects are best used in combination to delivery a more meaningful feedback experience.

The first of these structural framework aspects is "Describe procedure"; this can be used to detail a number of different elements of coursework; from how to implement a simple state machine using a case statement to the required steps within the IDE to debug the student's code correctly and effectively. Using the examples presented in the previous sentence showing the implementation of a state machine would require the student's code to be rewritten with a full explanation of why their code is incorrect and how the changes made to their code implement the state machine correctly. Likewise, for student submissions that did not execute, a debug process could be illustrated showing the debug process and a suitable narration, which, along with the required mouse clicks to access the appropriate menu options in the IDE, that would hopefully allow the student to solve a similar problem if encountered again. This ability to show a debug process in operation is a valuable process that merits a role out to all students as the ability to debug code is a valuable skill.

The second structural framework aspect is "Present concept"; this can be used to explain how collision detection works with regard to two sprites colliding. Again, examining the student's code and making the requisite changes while offering a suitable explanation of both the concept of collision detection and the code needed to implement it.

The third structural framework aspect is "Elaborate content"; Sugar et al. [25] suggest that this is the point at, which the screencast can "enrich learners'understanding and to encourage learners to consider other aspects of the process or concept". This aspect can be used to illustrate to the student how the concepts and techniques used to master the coursework can be embellished and reused in future courseworks and beyond.

The neatness and compactness of the actual code itself is an important aspect of any programming thus, the screencast gave the author the ability to highlight selected code within the Integrated Development Environment (IDE), in this case Microsoft Visual Studio, offering an audio narrative explaining clearly any deficient code and a visualisation of how the code could be reworked in order to make it neater and more efficient. Good examples of student work could also be highlighted and the student commended for its use.

Most of the screencasts were between 5 and 10 minutes in length depending on the game produced and the exhibited programming ability of the student, which is in keeping with the surveyed literature. Unlike the previous incarnation of the first trial [1] where he feedback screencasts were compressed into a .zip file and returned to the student via e-mail the decision was made to use the author's virtual learning environment (VLE), a version of Blackboard, to upload the video files directly to the student's secure storage area.

VI. Student Feedback

Having applied Sugar et al. [25] Screencast framework to the production of the screencasts the hope was that the student cohort would engage with the feedback screencasts, appreciating the audio-visual nature and the structured approach. Therefore it was heartening that the initial feedback from the students was, on the whole, positive and illuminating with regard to refining the screencast feedback process.

Although, comments were elicited from all (70) students in an informal manner, students were asked to complete a short Google Forms questionnaire, which, comprised of likert style questions and a short response question allowing them to give their initial impression of receiving feedback in this manner. The questions in the questionnaire were based on questions created by Ali [31]. Table IV shows the questions and the percentage, greater than or equal to 3 on the likert scale, responses. Of the 70 students who undertook the module there was 60 respondents to the questionnaire. As all students received both written feedback and feedback in the form of screencasts this allowed the students to compare and contrast the two forms of feedback proffering their thoughts.

Again, as with Law [1] the respondents overwhelming feeling was a sense of personalization and tailoring of feedback to their needs. Students were also receptive to the visual code analysis they received indicating that they understood more readily the need for well written, neat and compact code. Although, anecdotal, the quotes from students help to articulate their view of screencasts for feedback.

> *"The combination of seeing where I went wrong with Bobby's audio was very useful."*
>
> *"More personal feedback with clear direction on where I went wrong."*
>
> *"Seeing my game played by Bobby and with his comments really brought home to me where my interface was lacking."*

As with Law [1], the positive feedback suggests that the technique is worth persevering with and a further attempt will be made to hone the screencast framework prior to rolling out screencasting as a delivery mechanism for feedback. The Google Forms questionnaire will be restructured to aid with the capture of qualitative data.

VII. Issues

From the perspective of the lecturer there are some issues that need to be addressed. Seror [16], identifies that incorporating screencasts into his teaching "required adjustments to my regular feedback practices."

Firstly, the time taken to prepare the screencast feedback does not necessarily equate to the actual time of the screencast that the student will observe, which tallies with Mathieson [13], whom identifies this as an "important caveat" backing this up by noting that, during her trials, screencasts took approximately twice the time to produce. This is not, necessarily, due to the screencast being edited but the time taken to record the screencast itself. Kilickaya [14] suggests that the time taken to record screencast feedback may well be dependant on "the type of written work being marked as well as the comprehensibility of feedback."

Although, in Section V, a key piece of advise is to plan and prepare for the screencast by using some form of rubric, the application of this rubric can leave the recording having a staccato and unnatural feel. A solution to this is to pause the recording after each section and compose oneself before recording the next section. This will add time to the process but will prove worthwhile in the long term. The expectation would be, as noted by Atfield-cutts [11], that "the process sped up with practise to the point."

Secondly, the time consumed by planning, stopping short of scripting, the feedback. Implementing the screencast framework is not a quick process as this framework needs to be mapped against the coursework/assignment noting the key points of learning that should be fed back to the student, if and when appropriate. Again, this can be countered by the regular use of the technique and the fact that there may well be overlap between assignments, which, will lend itself to the re-use of some key points.

Thirdly, choosing a suitable location to record the screencasts is imperative as interruptions not only break the lecturers concentration but also can be inadvertently recorded thus, requiring the recording to be edited or, worse still, to be scrapped. A quiet location devoid of interruptions is not always possible in a busy University. It is not an insurmountable challenge but definitely something to be aware of prior to starting any recordings.

A fourth issue is the size of the recorded screencasts with regard to the required disk storage. The size is dependant on a number of factors including: video codec used, screen size being recorded, and resolution of recording. For example a screencast recorded using the H.264 video codec for YouTube with a definition of 720p, a resolution of 1280x720, 25 frames per second and lasting 5 minutes will require approximately 1.73 gigabytes of disk space. Thus, for a cohort of 70 students, approximately 121 gigabytes of disk storage would be required. This leads to a secondary issue with the delivery mechanism used for distributing the recordings to the students. Distribution by email can be a problem as there may be a restriction on the maximum file size that can be attached to an outgoing email. If this is the case then an alternative method will be required; this could be by uploading the file to a Managed Learning Environment (MLE). Cognisance should also be taken with regard to the time taken to return the feedback to the students [14] as it is not a trivial task to return sizeable video files.

All of the aforementioned issues are solvable with a bit of careful planning and preparation prior to embarking on the recording process.

VIII. Conclusion

Results from this second run of the project suggest that screencasts are, tentatively, potentially of benefit to students, but, may incur a time overhead for staff. From a student point of view, this would go along way to addressing the students perception of feedback as highlighted by the UK's National Student Survey.

Reflecting on the creation of the feedback screencasts, it is an interesting exercise to return to the five attributes of feedback, as defined by Race [12], and attempt to analyse, albeit subjectively, if screencast feedback can be thought of

TABLE IV. SURVEY RESULTS

Question	*≥3*
Did you feel receiving feedback through screencast videos helped you understand the programming techniques you implemented?	90%
Did you feel receiving feedback through screencast videos helped you improve your use of the C++ Standard Template Library?	76%
I found screencast videos helpful because I can replay the video at any time.	97%
I found screencast videos helpful because I can pause the video and reflect on how the code could be improved.	92%
I found screencast videos helpful as I understand where I have lost marks.	84%
The audio of the lecturer in the videos was clear.	95%
The language used giving the feedback was easy to understand.	93%
The Lecturer praised the positive aspects of my code.	87%
The feedback was supported by suggestions for improvement of my code.	88$
Watching screencast videos is time-consuming.	28%
I had difficulty loading the videos.	7%
I felt that receiving feedback through screencast videos engaged me actively in the process of code review and optimisation.	78%
I have a positive attitude toward receiving feedback through screencast videos	92%
Did you feel the feedback using screencast videos added a personal aspect?	94%

as improving these attributes. Again, this can be a time consuming exercise.

Timely feedback can be considered as a property of the turnaround time from student submission of coursework to the lecturer returning feedback to the student; to this end screencasting has no influence on this attribute.

Intimate and individual feedback is an interesting attribute; screencasts can help to achieve this attribute, especially for programming, as the student will receive feedback on their programming code, hearing and seeing the lecturer discuss various aspects of their game's code. Empowering feedback is a balance between providing positive feedback and being able to critic the student's work in such a manner that they feel engaged and enthused to progress and push forward. Screencasting feedback can provide the student with the necessary aural and visual cues to afford them the understanding of what is good with their work but also, in a positive manner, how their work can be improved. This is especially good for programming as it is important for students to understand that code that works can still be improved to make it more efficient and that this is a learning process and not a criticism. Open doors, not close them is a delicate area but with a judicious use of appropriate language and the correct vocal intonation the student can be presented with aural cues and, to a certain extent, visual cues that will allow them to synthesise the intended tone of the feedback.

Finally, Manageable, as noted by Race [12] has two aspects: the level of work involved for the lecturer and the volume of feedback given to the student. With regard to the level of work involved for the lecturer this may fluctuate depending on the cohort and the quality of their submissions, therefore, it is possible that it could add somewhat to the lecturers overhead for producing feedback. Hope for faculty would be that the process of creating the feedback screencasts would speed up with each iteration. However, for students, they should have a targeted and enhanced quality of feedback, which should not overburden them but provide the important aspects of the desired feedback they need to progress and improve.

Screencasts provide resource-rich feedback for students combining both narration and visual aspects to enrich and augment traditional feedback practices [16]. The increased feedback that can be crammed into a 5 minute screencast is more personal, clearer, less ambiguous than traditional written feedback and offers to show students "how to fix their own code or use a better technique, directly, without having to direct them to a generic example." [11], which would seem like a boon for the student. Although, oral feedback is given during lab sessions, this type of feedback is relevant in situ but, when the student refers back to this type of feedback it is entirely at the mercy of the student's ability to accurately record it. In contrast, the student can play and replay the video as many times as they like and the feedback will always be viewed as it was intended. The time to produce the screencasts varies by student submission but on the whole it was surprisingly quick in comparison to written feedback of the same depth.

IX. FUTURE WORK

The intention is to repeat the screencast feedback in the next academic year. The number of students undertaking the module will, again, be in the region of 60 students and should offer a suitable number for judging the timeliness of producing feedback screencasts. The hypothesis is that the experience from this first large scale implementation will lead to a more effective and quicker production process for each screencast and the students will benefit from clear, concise and helpful feedback.

The feedback screencasts will additionally be augmented by including webcam footage of the Lecturer, this will add back the visual and body language cues gained from face to face feedback [13], in the belief that it will "maximise the potential benefits of video feedback" [11].

The module is 12 weeks in duration and students will be asked to submit work at the end of week 8 and also at the end of week 12. Screencast feedback on their week 8 submission will be returned by week 10, which, should allow for the students to benefit from the feedback prior to their final submission in week 12 [15]. After receiving the feedback screencasts the students will be surveyed to ascertain a better representation of their feeling towards this feedback mechanism. Screencast feedback will be returned approximately 10 working days after week 12 submission and should serve to inform the students of their programming progress. The intention is to survey the students again at the end of the module in an attempt to better understand their opinion of screencasts as a means of delivering feedback. The survey will attempt to elicit the students perceptions of the screencast feedback based on the categories of engagement, quality and quantity of feedback, helpfulness and comparison to written feedback.

REFERENCES

[1] B. Law, "Screencasts : Enhancing Coursework Feedback for Game Programming Students," in The Twelfth International Multi-Conference on Computing in the Global Information Technology (ICCGI). Nice: The International Academy, Research and Industry Association (IARIA), 2017, pp. 17–21.

[2] N. U. of Students (NUS), "The nation student survey," 2015, [retrieved: July, 2017]. [Online]. Available: http://www.thestudentsurvey.com/

[3] S. A. Hope, "Making movies: The next big thing in feedback?" Bioscience Education, vol. 18, no. 1, 2011, pp. 1–14.

[4] K. Haxton and D. McGarvey, "Screencasting as a means of providing timely, general feedback on assessment," New Directions, vol. 7, 2011, pp. 18–21.

[5] N. U. of Students (NUS), "Nss 2015 national headlines," 2015, [retrieved: July, 2017]. [Online]. Available: http://www.thestudentsurvey.com/

[6] R. Law, "Using screencasts to enhance coursework feedback for game programming students," in Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ACM, 2013, pp. 329–329.

[7] R. Thompson and M. J. Lee, "Talking with students through screencasting: Experimentations with video feedback to improve student learning," The Journal of Interactive Technology and Pedagogy, vol. 1, no. 1, 2012.

[8] M. Robinson, B. Loch, and T. Croft, "Student perceptions of screencast feedback on mathematics assessment," International Journal of Research in Undergraduate Mathematics Education, vol. 1, no. 3, 2015, pp. 363–385.

[9] K. Edwards, A.-F. Dujardin, and N. Williams, "Screencast feedback for essays on a distance learning ma in professional communication," Journal of Academic Writing, vol. 2, no. 1, 2012, pp. 95–126.

[10] G. Stieglitz, "Screencasting: Informing students, shaping instruction," UAE Journal of Educational Technology and eLearning, vol. 4, no. 1, 2013, pp. 58–62.

[11] S. Atfield-cutts and M. Coles, "Blended Feedback II : Video screen capture assessment feedback for individual students , as a matter of course , on an undergraduate computer programming unit," 2013. [Online]. Available: http://eprints.bournemouth.ac.uk/23813/

[12] P. Race, "Using feedback to help students to learn," HEA, York, 2001.

[13] K. Mathieson, "Exploring student perceptions of audiovisual feedback via screencasting in online courses," American Journal of Distance Education, vol. 26, no. 3, 2012, pp. 143–156.

[14] F. Kiliçkaya, "Use of Screencasting for Delivering Lectures and Providing Feedback in Educational Contexts: Issues and Implications." Online Submission, 2016. [Online]. Available: https://eric.ed.gov/?id=ED574888

[15] C. Evans, "Making sense of assessment feedback in higher education," Review of Educational Research, vol. 83, no. 1, 2013, pp. 70–120. [Online]. Available: http://dx.doi.org/10.3102/0034654312474350

[16] J. Seror, "Show me! enhanced feedback through screencasting technology." TESL Canada Journal, vol. 30, no. 1, 2012, pp. 104–116.

[17] M. Henderson and M. Phillips, "Video-based feedback on student assessment: scarily personal." Australasian Journal of Educational Technology, vol. 31, no. 1, 2015, pp. 51–66.

[18] F. Harper, H. Green, and M. Fernandez-Toro, "Using screencasts in the teaching of modern languages: investigating the use of jing® in feedback on written assignments," The Language Learning Journal, 2015, pp. 1–18.

[19] R. Galanos, W. Brand, S. Sridhara, M. Zamansky, and E. Zayas, "Technology we can't live without!: revisited," in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM, 2017, pp. 659–660.

[20] J. West and W. Turner, "Enhancing the assessment experience: improving student perceptions, engagement and understanding using online video feedback," Innovations in Education and Teaching International, 2015, pp. 1–11.

[21] P. Marriott and L. K. Teoh, "Using screencasts to enhance assessment feedback: Students' perceptions and preferences," Accounting Education, vol. 21, 2012, pp. 583–598.

[22] P. Mathisen, "Video feedback in higher education–a contribution to improving the quality of written feedback," Nordic Journal of Digital Literacy, vol. 7, no. 02, 2012, pp. 97–113.

[23] P. OMalley, "Screencasting and a tablet pc–an indispensable technology combination for physical science teaching and feedback in higher and further education," in Aiming for excellence in STEM learning and teaching: Proceedings of the Higher Education Academy's First Annual Learning and Teaching STEM Conference, 2012.

[24] C. Chewar and S. J. Matthews, "Lights, camera, action!: video deliverables for programming projects," Journal of Computing Sciences in Colleges, vol. 31, no. 3, 2016, pp. 8–17.

[25] W. Sugar, A. Brown, and K. Luterbach, "Examining the anatomy of a screencast: Uncovering common elements and instructional strategies," International Review of Research in Open and Distance Learning, vol. 11, no. 3, 2010, pp. 1–20. [Online]. Available: http://www.irrodl.org/index.php/irrodl/article/view/851

[26] P. Smith, "Screencasting as a means of enhancing the student learning experience," Learning and Teaching in Action, 2014, p. 59.

[27] D. Wolff-Hilliard and B. Baethe, "Using digital and audio annotations to reinvent critical feedback with online adult students," International Journal for Professional Educators, 2014, p. 40.

[28] S. Mohorovicic, "Creation and use of screencasts in higher education," in MIPRO, 2012 Proceedings of the 35th International Convention. IEEE, 2012, pp. 1293–1298.

[29] S. Mohorovičić and E. Tijan, "Using Screencasts in Computer Programming Courses," in Proceedings of the 22nd EAEEIE Annual Conference, Maribor, 2011, pp. 220–225. [Online]. Available: http://www.eaeeie2011.uni-mb.si/eaeeie2011_submission_48.pdf

[30] L. A. Jones, "Losing the red pen: Video grading feedback in distance and blended learning writing courses," Association Supporting Computer Users in Education Our Second Quarter Century of Resource Sharing, 2014, p. 54.

[31] A. D. Ali, "Effectiveness of using screencast feedback on efl students writing and perception," English Language Teaching, vol. 9, no. 8, 2016, p. 106.